НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №8

з дисципліни **«**Програмування паралельних комп’ютерних систем**»**

Виконала:

студентка 3 курсу

ФІОТ гр. ІО-21

Заєць С. Ю.

Перевірив:

Корочкін О. В.

Київ – 2015 р.

Тема: Програмування для комп’ютерних систем з локальною пам’яттю. Бібліотека MPI.

Розробити програму для розв’язання в ПКС із ЛП математичної задачі:

MA = MB∙MC + α∙MK

Бібліотека програмування: MPI



Структурна схема ПКС

**Виконання роботи:**

**Етап 1. Побудова паралельного алгоритму**

1. MAH = MB∙MCH + α∙MMH

**Етап 2. Розроблення алгоритмів роботи кожного процесу**

**Задача Т0**

1. Введення α, MB, MC, MM
2. **Передати** α, MB, MCH, MMH задачам Т1-Т3
3. **Передати** α, MB, MC4H, MM4H задачі Т4
4. Обчислити MAH = MB∙MCH + α∙MMH
5. **Прийняти** MAH від задач Т1-Т3
6. **Прийняти** MA4H від задачі Т4
7. Вивести МА

**Задача Т1-Т3**

1. **Прийняти** α, MB, MCH, MMH від задачі Т0
2. Обчислити MAH = MB∙MCH + α∙MMH
3. **Передати** MAH задачі Т0

**Задача Т4**

1. **Прийняти** α, MB, MC4H, MM4H від задачі Т0
2. **Передати** α, MB, MC4H, MM4H задачам Т5-Т7
3. Обчислити MAH = MB∙MCH + α∙MMH
4. Прийняти MAH від задач Т5-Т7
5. **Передати** MA4H задачі Т0

**Задача Т5-Т7**

1. **Прийняти** α, MB, MCH, MMH від задачі Т4
2. Обчислити MAH = MB∙MCH + α∙MMH
3. **Передати** MAH задачі Т4

Етап 4. Розроблення програми

1. import mpi.Graphcomm;
2. import mpi.Intracomm;
3. import mpi.MPI;
4. public class Executor {
5. public static int N;
6. public static int P;
7. public static int H;
8. public static void main(String[] args) {
9. P = Integer.parseInt(args[1]);
10. N = Integer.parseInt(args[3]);
11. H = N / P;
12. MPI.Init(args);
13. int[] index = new int[] { 4, 5, 6, 7, 11, 12, 13, 14 };
14. int[] edges = new int[] { 1, 2, 3, 4, 0, 0, 0, 0, 5, 6, 7, 4, 4, 4 };
15. Graphcomm graph = MPI.COMM\_WORLD.Create\_graph(index, edges, false);
17. grapfTopoTest(graph);
18. int[][] sendBuf = new int[8 \* N + 2 \* (H \* 8) + 8][N];
19. int[][] recvBuf = new int[N + H + H + 1][N];
20. int[][] MC\_send = new int[N][N];
21. int[][] MM\_send = new int[N][N];
22. int[][] MB\_send = new int[N][N];
23. int alfa;
24. int[][] MC\_recv = new int[H][N];
25. int[][] MM\_recv = new int[H][N];
26. int[][] MB\_recv = new int[N][N];
27. int[][] MA\_send = new int[H][N];
28. int[][] MA\_resv = new int[N][N];
29. //1. Введення α, MB, MC, MM
30. if (graph.Rank() == 0) {
31. for (int i = 0; i < N; i++) {
32. for (int j = 0; j < N; j++) {
33. MB\_send[i][j] = 1;
34. MC\_send[i][j] = 1;
35. MM\_send[i][j] = 1;
36. }
37. }
38. int i = 0;
39. int g = 0;
40. int y = 0;
41. for (int j = 0; j < graph.Size(); j++) {
42. // write MB
43. for (int j2 = 0; j2 < N; j2++, i++) {
44. for (int k = 0; k < N; k++) {
45. sendBuf[i][k] = MB\_send[j2][k];
46. }
47. }
48. // write mc
49. for (; g < H \* (j + 1); g++, i++) {
50. for (int j2 = 0; j2 < MC\_send.length; j2++) {
51. sendBuf[i][j2] = MC\_send[g][j2];
52. }
53. }
54. // write mm
55. for (; y < H \* (j + 1); y++, i++) {
56. for (int j2 = 0; j2 < MM\_send.length; j2++) {
57. sendBuf[i][j2] = MM\_send[y][j2];
58. }
59. }
60. sendBuf[i][0] = 1;
61. i++;
62. }
64. }
65. // 2. Передати/Передати α, MB, MCH, MMH задачам Т1-Т3
66. // 3. Передати/Передати α, MB, MC4H, MM4H задачі Т4
67. graph.Scatter(sendBuf, 0, N + 2 \* H + 1, MPI.OBJECT, recvBuf, 0, N + 2
68. \* H + 1, MPI.OBJECT, 0);
69. int i = 0;
70. // get MB
71. for (; i < MB\_recv.length; i++) {
72. for (int j = 0; j < MB\_recv[i].length; j++) {
73. MB\_recv[i][j] = recvBuf[i][j];
74. }
75. }
76. // getMM
77. for (int j = 0; j < MC\_recv.length; j++, i++) {
78. for (int k = 0; k < MC\_recv[j].length; k++) {
79. MC\_recv[j][k] = recvBuf[i][k];
80. }
81. }
82. // getMM
83. for (int j = 0; j < MM\_recv.length; j++, i++) {
84. for (int k = 0; k < MM\_recv[j].length; k++) {
85. MM\_recv[j][k] = recvBuf[i][k];
86. }
87. }
88. //input alpha
89. alfa = recvBuf[i][0];
90. for (int j = 0; j < H; j++) {
91. for (int k = 0; k < N; k++) {
92. MA\_send[j][k] = 0;
93. for (int m = 0; m < N; m++) {
94. MA\_send[j][k] += MC\_recv[j][m] \* MB\_recv[m][k];
95. }
96. MA\_send[j][k] += MM\_recv[j][k] \* alfa;
97. }
98. }
100. // 5. Прийняти/Передати MAH від задач Т1-Т3
101. // 6. Прийняти/Передати MA4H від задачі Т4
102. graph.Gather(MA\_send, 0, H, MPI.OBJECT, MA\_resv, 0, H, MPI.OBJECT, 0);
103. if (graph.Rank() == 0) {
104. System.out.println("Result");
105. //7. Вивести МА
106. matrixOutput(MA\_resv);
107. }
108. MPI.Finalize();
109. }
110. public static void grapfTopoTest(Graphcomm comm) {
111. if (comm.Rank() == 0) {
112. for (int rank = 0; rank < 8; rank++) {
113. System.out.println("I am node " + rank + " my neighbors is:");
114. for (int nbrs : comm.Neighbours(rank)) {
115. System.out.print(nbrs + ", ");
116. }
117. System.out.println();
118. System.out.println();
119. }
120. }
121. }
122. public static void matrixOutput(int[][] array) {
123. for (int i = 0; i < array.length; i++) {
124. for (int j = 0; j < array[i].length; j++) {
125. System.out.print(array[i][j] + ", ");
126. }
127. System.out.println();
128. }
129. System.out.println();
130. }
131. }